



Red Hat Advanced Cluster Management for Kubernetes 2.6

Clusters

[Read more about cluster lifecycle.](#)

Red Hat Advanced Cluster Management for Kubernetes 2.6 Clusters

[Read more about cluster lifecycle.](#)

Legal Notice

Copyright © 2023 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

XFS[®] is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Abstract

Read more about cluster lifecycle.

Table of Contents

CHAPTER 1. CLUSTER LIFECYCLE ARCHITECTURE	3
1.1. RELEASE IMAGES	3
1.1.1. Creating a release image to deploy a cluster on a different architecture	5
1.1.2. Maintaining a custom list of release images when connected	6
1.1.3. Synchronizing available release images	8
1.2. MODIFYING THE KLUSTERLET ADD-ONS SETTINGS OF YOUR CLUSTER	9
1.2.1. Description of klusterlet add-ons settings	10
1.2.2. Modify using the console on the hub cluster	10
1.2.3. Modify using the command line on the hub cluster	11

CHAPTER 1. CLUSTER LIFECYCLE ARCHITECTURE

Red Hat Advanced Cluster Management for Kubernetes has two main types of clusters: *hub clusters* and *managed clusters*, which are created and managed by the multicluster engine for Kubernetes operator. See the [The multicluster engine for Kubernetes overview](#) that covers everything from installation to troubleshooting.

To continue adding and managing clusters in the cluster lifecycle phase, you must start with [The multicluster engine operator cluster lifecycle overview](#) and all the documentation for that operator.

Your cluster is created by using the Red Hat OpenShift Container Platform cluster installer with the Hive resource. You can find more information about the process of installing clusters at [OpenShift Container Platform installation overview](#) in the OpenShift Container Platform documentation.

The components of the cluster lifecycle management architecture are included in the [Cluster lifecycle architecture](#).

The hub cluster is the main cluster with Red Hat Advanced Cluster Management for Kubernetes installed. You can create, manage, and monitor other Kubernetes clusters with the hub cluster. The managed clusters are Kubernetes clusters that are managed by the hub cluster. You can create clusters by using the Red Hat Advanced Cluster Management hub cluster, while you can also import existing clusters to be managed by the hub cluster.

Release images are the version of OpenShift Container Platform that you use when you create a cluster. For clusters that are created using Red Hat Advanced Cluster Management, you can enable automatic upgrading of your release images. For more information about release images in Red Hat Advanced Cluster Management, see [Release images](#).

1.1. RELEASE IMAGES

When you create a cluster on a provider by using Red Hat Advanced Cluster Management for Kubernetes, you must specify a release image to use for the new cluster. The release image specifies which version of Red Hat OpenShift Container Platform is used to build the cluster.

The files that reference the release images are YAML files that are maintained in the **acm-hive-openshift-releases** GitHub repository. Red Hat Advanced Cluster Management uses those files to create the list of the available release images in the console. This includes the latest fast channel images from OpenShift Container Platform. The console only displays the latest release images for the three latest versions of OpenShift Container Platform. For example, you might see the following release images displayed in the console options:

- `quay.io/openshift-release-dev/ocp-release:4.6.23-x86_64`
- `quay.io/openshift-release-dev/ocp-release:4.10.1-x86_64`

Note: Only release images with the label of: **visible: 'true'** are available to select when creating clusters in the console. An example of this label in a **ClusterImageSet** resource is provided in the following content:

```
apiVersion: config.openshift.io/v1
kind: ClusterImageSet
metadata:
  labels:
    channel: fast
    visible: 'true'
```

```
name: img4.10.1-x86-64-appsub
spec:
  releaseImage: quay.io/openshift-release-dev/ocp-release:4.10.1-x86_64
```

Additional release images are stored, but are not visible in the console. To view all of the available release images, run **kubectrl get clusterimageset** in your CLI. Only the latest versions are in the console to encourage the creation of clusters with the latest release images. In some cases, you might need to create a cluster that is a specific version, which is why the older versions are available. Red Hat Advanced Cluster Management uses those files to create the list of the available release images in the console. This includes the latest fast channel images from OpenShift Container Platform.

The repository contains the **clusterImageSets** directory and the **subscription** directory, which are the directories that you use when working with the release images.

The **clusterImageSets** directory contains the following directories:

- **Fast**: Contains files that reference the latest versions of the release images for each OpenShift Container Platform version that is supported. The release images in this folder are tested, verified, and supported.
- **Releases**: Contains files that reference all of the release images for each OpenShift Container Platform version (stable, fast, and candidate channels) **Note**: These releases have not all been tested and determined to be stable.
- **Stable**: Contains files that reference the latest two stable versions of the release images for each OpenShift Container Platform version that is supported.

Note: By default, the current list of release images is updated one time an hour. After upgrading the product, it may take up to an hour for the list to reflect the recommended release image versions for the new version of the product.

You can curate your own **ClusterImageSets** in three ways:

The first step for any of the three ways is to disable the included subscription that automatically updates the latest fast channel images. The automatic curation of the latest fast **ClusterImageSets** can be disabled by using an installer parameter on the **multiclusterhub** resource. By toggling the **spec.disableUpdateClusterImageSets** parameter between **true** and **false**, the subscription installed with Red Hat Advanced Cluster Management is disabled or enabled, respectively. If you want to curate your own images, set the **spec.disableUpdateClusterImageSets** to **true** to disable the subscription.

Option 1: Specify the image reference for the specific **ClusterImageSet** that you want to use in the console when creating a cluster. Each new entry you specify persists and is available for all future cluster provisions. An example of an entry is: **quay.io/openshift-release-dev/ocp-release:4.6.8-x86_64**.

Option 2: Manually create and apply a **ClusterImageSets** YAML file from the **acm-hive-openshift-releases** GitHub repository.

Option 3: Follow the **README.md** in the **acm-hive-openshift-releases** GitHub repository to enable automatic updates of **ClusterImageSets** from a forked GitHub repository.

The **subscription** directory contains files that specify where the list of release images is pulled from.

The default release images for Red Hat Advanced Cluster Management are provided in a Quay.io directory.

The images are referenced by the files in the [acm-hive-openshift-releases GitHub repository for release 2.5](#).

1.1.1. Creating a release image to deploy a cluster on a different architecture

You can create a cluster on an architecture that is different from the architecture of the hub cluster by manually creating a release image that contains the files for both architectures.

For example, you might need to create an **x86_64** cluster from a hub cluster that is running on the **ppc64le**, **aarch64**, or **s390x** architecture. If you create the release image with both sets of files, the cluster creation succeeds because the new release image enables the OpenShift Container Platform release registry to provide a multi-architecture image manifest.

OpenShift Container Platform 4.11 and later supports multiple architectures by default. You can use the following **clusterImageSet** to provision a cluster:

```
apiVersion: hive.openshift.io/v1
kind: ClusterImageSet
metadata:
  labels:
    channel: fast
    visible: 'true'
  name: img4.12.0-multi-appsub
spec:
  releaseImage: quay.io/openshift-release-dev/ocp-release:4.12.0-multi
```

To create the release image for OpenShift Container Platform images that do not support multiple architectures, complete steps similar to the following example for your architecture type:

1. From the [OpenShift Container Platform release registry](#), create a [manifest list](#) that includes **x86_64**, **s390x**, **aarch64**, and **ppc64le** release images.
 - a. Pull the manifest lists for both architectures in your environment from the [Quay repository](#) using the following example commands:

```
podman pull quay.io/openshift-release-dev/ocp-release:4.10.1-x86_64
podman pull quay.io/openshift-release-dev/ocp-release:4.10.1-ppc64le
podman pull quay.io/openshift-release-dev/ocp-release:4.10.1-s390x
podman pull quay.io/openshift-release-dev/ocp-release:4.10.1-aarch64
```

- b. Log in to your private repository where you maintain your images:

```
podman login <private-repo>
```

Replace **private-repo** with the path to your repository.

- c. Add the release image manifest to your private repository by running the following commands that apply to your environment:

```
podman push quay.io/openshift-release-dev/ocp-release:4.10.1-x86_64 <private-repo>/ocp-release:4.10.1-x86_64
podman push quay.io/openshift-release-dev/ocp-release:4.10.1-ppc64le <private-repo>/ocp-release:4.10.1-ppc64le
podman push quay.io/openshift-release-dev/ocp-release:4.10.1-s390x <private-repo>/ocp-release:4.10.1-s390x
podman push quay.io/openshift-release-dev/ocp-release:4.10.1-aarch64 <private-repo>/ocp-release:4.10.1-aarch64
```

Replace **private-repo** with the path to your repository.

- d. Create a manifest for the new information:

```
podman manifest create mymanifest
```

- e. Add references to both release images to the manifest list:

```
podman manifest add mymanifest <private-repo>/ocp-release:4.10.1-x86_64
podman manifest add mymanifest <private-repo>/ocp-release:4.10.1-ppc64le
podman manifest add mymanifest <private-repo>/ocp-release:4.10.1-s390x
podman manifest add mymanifest <private-repo>/ocp-release:4.10.1-aarch64
```

Replace **private-repo** with the path to your repository.

- f. Merge the list in your manifest list with the existing manifest:

```
podman manifest push mymanifest docker://<private-repo>/ocp-release:4.10.1
```

Replace **private-repo** with the path to your repository.

2. On the hub cluster, create a release image that references the manifest in your repository.

- a. Create a YAML file that contains information that is similar to the following example:

```
apiVersion: hive.openshift.io/v1
kind: ClusterImageSet
metadata:
  labels:
    channel: fast
    visible: "true"
  name: img4.10.1-appsub
spec:
  releaseImage: <private-repo>/ocp-release:4.10.1
```

Replace **private-repo** with the path to your repository.

- b. Run the following command on your hub cluster to apply the changes:

```
oc apply -f <file-name>.yaml
```

Replace **file-name** with the name of the YAML file that you just created.

3. Select the new release image when you create your OpenShift Container Platform cluster.
4. If you deploy the managed cluster using the Red Hat Advanced Cluster Management console, specify the architecture for the managed cluster in the *Architecture* field during the cluster creation process.

The creation process uses the merged release images to create the cluster.

1.1.2. Maintaining a custom list of release images when connected

You might want to ensure that you use the same release image for all of your clusters. To simplify, you can create your own custom list of release images that are available when creating a cluster. Complete the following steps to manage your available release images:

1. If the installer-managed **acm-hive-openshift-releases** subscription is enabled, disable it by setting the value of **disableUpdateClusterImageSets** to **true** in the **multiclusterhub** resource.
2. Fork the [acm-hive-openshift-releases GitHub repository 2.5 branch](#).
3. Update the **./subscribe/channel.yaml** file by changing the **spec: pathname** to access your the GitHub name for your forked repository, instead of **stolostron**. This step specifies where the hub cluster retrieves the release images. Your updated content should look similar to the following example:

```
spec:
  type: Git
  pathname: https://github.com/<forked_content>/acm-hive-openshift-releases.git
```

Replace **forked_content** with the path to your forked repository.

4. Add the YAML files for the images that you want available when you create a cluster by using the Red Hat Advanced Cluster Management for Kubernetes console to the **./clusterImageSets/stable/*** or **./clusterImageSets/fast/*** directory.
Tip: You can retrieve the available YAML files from the main repository by merging changes into your forked repository.
5. Commit and merge your changes to your forked repository.
6. To synchronize your list of fast release images after you have cloned the **acm-hive-openshift-releases** repository, enter the following command to update the fast images:

```
make subscribe-fast
```

Note: You can only run this **make** command when you are using the Linux or MacOS operating system.

After running this command, the list of available fast release images updates with the currently available images in about one minute.

7. By default, only the fast images are listed. To synchronize and display the stable release images, enter the following command:

```
make subscribe-stable
```

Note: You can only run this **make** command when you are using the Linux or MacOS operating system.

After running this command, the list of available stable release images updates with the currently available images in about 1 minute.

8. By default, Red Hat Advanced Cluster Management pre-loads a few ClusterImageSets. You can use the following commands to list what is available and remove the defaults.

```
oc get clusterImageSets
oc delete clusterImageSet <clusterImageSet_NAME>
```

Note: If you have not disabled the installer-managed automatic updates of the **ClusterImageSets** by setting the value of **disableUpdateClusterImageSets** to **true** in the **multiclusterhub** resource, any images that you delete are recreated automatically.

- View the list of currently available release images in the Red Hat Advanced Cluster Management console when you are creating a cluster.

1.1.3. Synchronizing available release images

The release images are updated frequently, so you might want to synchronize the list of release images to ensure that you can select the latest available versions. The release images are available in the [acm-hive-openshift-releases](#) GitHub repository for release 2.5.

There are three levels of stability of the release images:

Table 1.1. Stability levels of release images

Category	Description
stable	Fully tested images that are confirmed to install and build clusters correctly.
fast	Partially tested, but likely less stable than a stable version.
candidate	Not tested, but the most current image. Might have some bugs.

Complete the following steps to refresh the list:

- If the installer-managed **acm-hive-openshift-releases** subscription is enabled, disable the subscription by setting the value of **disableUpdateClusterImageSets** to **true** in the **multiclusterhub** resource.
- Clone the [acm-hive-openshift-releases](#) GitHub repository for release 2.5.
- Remove the subscription by entering a command that is similar to the following command:

```
oc delete -f subscribe/subscription-fast
```

- Connect to the stable release images and synchronize your Red Hat Advanced Cluster Management for Kubernetes hub cluster by entering the following command:

```
make subscribe-stable
```

Note: You can only run this **make** command when you are using the Linux or MacOS operating system.

After about one minute, the latest list of **stable** release images is available.

- To synchronize and display the fast release images, enter the following command:

```
make subscribe-fast
```

Note: You can only run this **make** command when you are using the Linux or MacOS operating system.

About one minute after running the command, the list of available **stable** and **fast** release images updates with the currently available images.

- To synchronize and display the **candidate** release images, enter the following command:

```
make subscribe-candidate
```

Note: You can only run this **make** command when you are using the Linux or MacOS operating system.

About one minute after running the command, the list of available **stable**, **fast**, and **candidate** release images updates with the currently available images.

5. View the list of currently available release images in the Red Hat Advanced Cluster Management console when you are creating a cluster.
6. You can unsubscribe from any of these channels to stop viewing the updates by entering a command in the following format:

```
oc delete -f subscribe/subscription-fast
```

1.2. MODIFYING THE KLUSTERLET ADD-ONS SETTINGS OF YOUR CLUSTER

You can modify the settings of **KlusterletAddonConfig** to change your configuration using the hub cluster.

The **KlusterletAddonConfig** controller manages the functions that are enabled and disabled according to the settings in the **klusterletaddonconfigs.agent.open-cluster-management.io** Kubernetes resource. View the following example of the **KlusterletAddonConfig**:

```
apiVersion: agent.open-cluster-management.io/v1
kind: KlusterletAddonConfig
metadata:
  name: <cluster-name>
  namespace: <cluster-name>
spec:
  clusterName: <cluster-name>
  clusterNamespace: <cluster-name>
  clusterLabels:
    cloud: auto-detect
    vendor: auto-detect
  applicationManager:
    enabled: true
  certPolicyController:
    enabled: true
  iamPolicyController:
    enabled: true
  policyController:
    enabled: true
```

```
searchCollector:
  enabled: false
  version: 2.5.0
```

1.2.1. Description of klusterlet add-ons settings

The following settings can be updated in the **klusterletaddonconfigs.agent.open-cluster-management.io** Kubernetes resource:

Table 1.2. Table list of klusterlet add-ons settings

Setting name	Value	Description
applicationmanager	true or false	This controller manages the application subscription lifecycle on the managed cluster.
certPolicyController	true or false	This controller enforces certificate-based policies on the managed cluster.
iamPolicyController	true or false	This controller enforces the IAM-based policy lifecycle on the managed cluster.
policyController	true or false	This controller enforces all other policy rules on the managed cluster.
searchCollector	true or false	This controller is used to periodically push the resource index data back to the hub cluster.

1.2.2. Modify using the console on the hub cluster

You can modify the settings of the **klusterletaddonconfigs.agent.open-cluster-management.io** resource by using the hub cluster. Complete the following steps to change the settings:

1. Log in to the Red Hat Advanced Cluster Management for Kubernetes console of the hub cluster.
2. From the header menu of the hub cluster console, select the **Search** icon.
3. In the search parameters, enter the following value: **kind:klusterletaddonconfigs**
4. Select the endpoint resource that you want to update.
5. Find the **spec** section and select **Edit** to edit the content.
6. Modify your settings.
7. Select **Save** to apply your changes.

1.2.3. Modify using the command line on the hub cluster

You must have access to the **cluster-name** namespace to modify your settings by using the hub cluster. Complete the following steps:

1. Log in to the hub cluster.
2. Enter the following command to edit the resource:

```
kubectl edit klusterletaddonconfigs.agent.open-cluster-management.io <cluster-name> -n <cluster-name>
```

3. Find the **spec** section.
4. Modify your settings, as necessary.